Artificial Intelligence Nanodegree
Assignment 2 – Advanced Game Playing – Building a Game Playing Agent
Charles Ocheret
February 19, 2017

A variety custom heuristic functions were developed, including one parametric one. Three are described here:

1. Heuristic A – Counts the number of moves available to each player in exactly 1 and 2 moves and combines them in a fashion meant to weigh the importance of the type of move. Given:
    - P1 = number of open spaces for player to move to in exactly 1 move
    - P2 = number of open spaces for player to move to in exactly 2 moves
    - O1 = number of open spaces for opponent to move to in exactly 1 move
    - O2 = number of open spaces for opponent to move to in exactly 2 moves

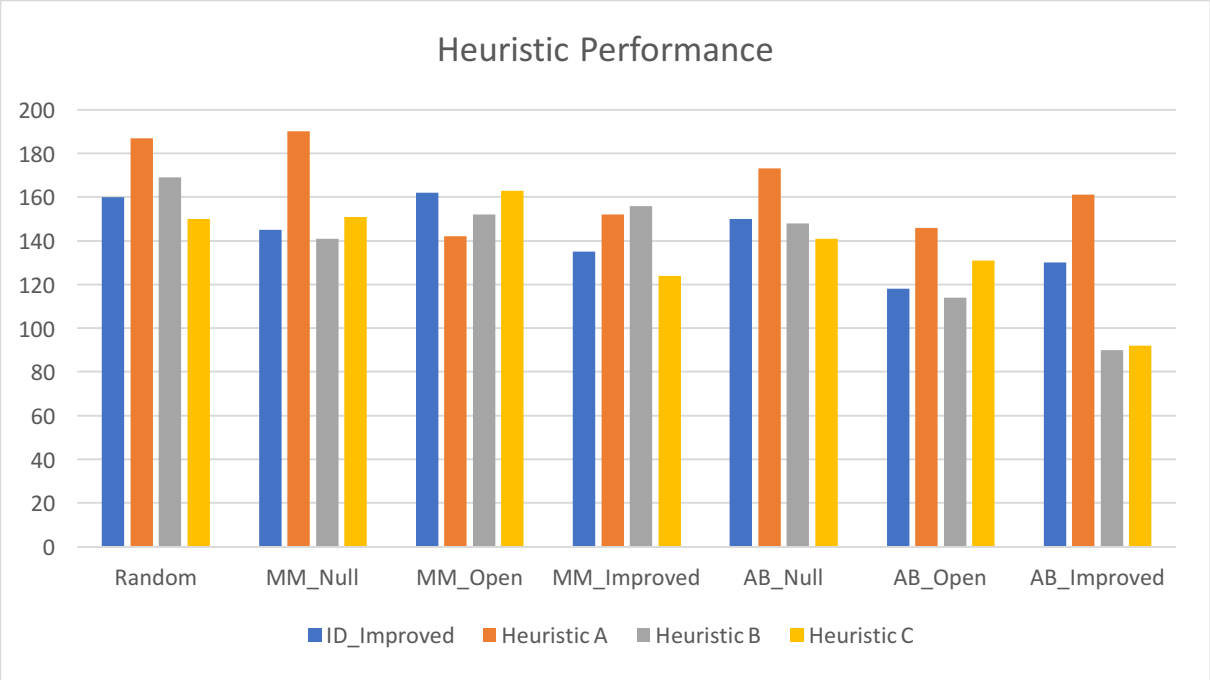
    Compute a heuristic score as (P1 + F1 * P2) – (O1 + F2 * P2) where F1 and F2 are weights intended to reduce the importance of moves further in the future. After a number of trials, F1 and F2 were found to perform optimally where F1=0.9 and F2=0.8. This heuristic is intended to reflect how much room each player has to expand into beyond just the next move. However, it discounts the importance of later moves based on the increased uncertainty associated with them.
2. Heuristic B – Computes the difference between the number of legal moves available to the player and the opponent. Added to this difference is the number of legal moves in common between the players. This is intended to reflect the advantage the current player has by being able to move into spots that might have been available to the opponent.
3. Heuristic C – Computes the weighted difference between the number of legal moves available to the player and the opponent. A factor is multiplied by the opponent's count to reduce the importance of their moves since not all moves will be possible once an actual choice is made by the player.

The following table summarizes the relative success rates of each heuristic and compares them to the ID_Improved heuristic:

Wins and Percentages for 200 Games

	ID_Improved		**Heuristic A**		**Heuristic B**		**Heuristic C**	
vs	Wins	% Wins	Wins	% Wins	Wins	% Wins	Wins	% Wins
Random	160	80.00%	187	93.50%	169	84.50%	150	75.00%
MM_Null	145	72.50%	190	95.00%	141	70.50%	151	75.50%
MM_Open	162	81.00%	142	71.00%	152	76.00%	163	81.50%
MM_Improved	135	67.50%	152	76.00%	156	78.00%	124	62.00%
AB_Null	150	75.00%	173	86.50%	148	74.00%	141	70.50%
AB_Open	118	59.00%	146	73.00%	114	57.00%	131	65.50%
AB_Improved	130	65.00%	161	80.50%	90	45.00%	92	46.00%
TOTAL	1000	**71.43%**	1151	**82.21%**	970	**69.29%**	952	**68.00%**

This information is easily represented in the following chart:



Heuristic A was selected for several reasons.

- It had the highest overall success rate at 82%. The next closest was 71%.
- The only case where it performed worse than the other heuristics was during the MM_Open case. The "Open" heuristic has the advantage of extremely low computational overhead, resulting in more plies examined. I believe that a more highly optimized implementation of Heuristic A might be able to overcome this.
- Even in the single case where Heuristic A was not the best performer, it still won 71% of time time, which will still make me money at the Isolation tables in Las Vegas. ;-)

Several other more promising variants of this heuristic were attempted. For example, a generalized heuristic that examined a weighted sum of the legal moves counts up to 3 moves ahead. However, the additional overhead appeared to reduce amount of work that could be done during iterative deepening and resulted in lower success rates. Heuristic A appears to have a good balance of being a beneficial heuristic with low enough computational overhead for its value to be leveraged well.

There was quite a bit of variability in the success rates between different runs in all cases. For this reason I increased the number of games run in each tournament to attempt to arrive at more representative results.